AFRL-IF-RS-TR-2004-180
Final Technical Report
June 2004

FROM LANGUAGE TO KNOWLEDGE: STARTING HAWK

Massachusetts Institute of Technology

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J885

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-180 has been reviewed and is approved for publication

APPROVED: /s/

JOHN SPINA
Project Engineer

FOR THE DIRECTOR: /s/

JAMES A. COLLINS, Acting Chief
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JUNE 2004	Final Jun 00 – Sep 03

4. TITLE AND SUBTITLE
FROM LANGUAGE TO KNOWLEDGE: STARTING HAWK

5. FUNDING NUMBERS
C - F30602-00-1-0545
PE - 62301E
PR - RKFM
TA - 00
WU - 04

6. AUTHOR(S)
Boris Katz,
Gary Borchardt, and
Sue Felshin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology
545 Technology Square, NE43-824
Cambridge Massachusetts 02139

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency AFRL/IFTB
3701 North Fairfax Drive 525 Brooks Road
Arlington Virginia 22203-1714 Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2004-180

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: John Spina/IFTB/(315) 330-4032/ John.Spina@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

This report describes work completed by the MIT Computer Science and Artificial Intelligence Laboratory in support of DARPA's Rapid Knowledge Formation (RKF) program over the period from July 2000 to September 2003. The primary focus of the RKF program is to develop new technology to automate the task of transforming raw human-understandable information into encoded, machine-understandable information. The project described in this report addresses a central subtask of this task: converting natural language text into an encoded representation that can support computer inference. The technical approach taken in this effort is based on two key insights: First, we can make the translation task manageable by breaking it into successive stages of isolating information, then standardizing it, then encoding it, with each stage facilitated by proven components of natural language processing technology. Second, we can gain leverage during the translation process by exploiting human interaction at a number of distinct points along the way.

14. SUBJECT TERMS
Knowledge base, Natural Language Processing, Knowledge Representation, Computer Inference, Reasoning

15. NUMBER OF PAGES
39

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

1. Overview

This report describes work completed by the MIT Computer Science and Artificial Intelligence Laboratory in support of DARPA's Rapid Knowledge Formation (RKF) program over the period from July 2000 to September 2003. The primary focus of the RKF program is to develop new technology to automate the task of transforming raw human-understandable information into encoded, machine-understandable information. The project described in this report addresses a central subtask of this task: converting natural language text into an encoded representation that can support computer inference. The technical approach taken in this effort is based on two key insights:

- First, we can make the translation task manageable by breaking it into successive stages of isolating information, then standardizing it, then encoding it, with each stage facilitated by proven components of natural language processing technology.
- Second, we can gain leverage during the translation process by exploiting human interaction at a number of distinct points along the way.

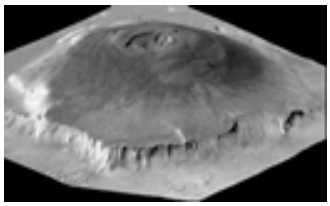
Supporting this effort is a key technology component grounded in sentence-level natural language processing. Whereas research in natural language processing has encountered significant difficulties in handling larger units of discourse, much progress has been made in mapping natural language phrases and sentences into sets of underlying semantic relationships that can be usefully manipulated by computers. Thus, this project takes the position of treating sentence-level natural language as itself a suitable representation for information content. This position is embodied in the notion of *natural language annotation*, whereby natural language phrases and sentences are used to organize and describe the content of arbitrary multi-media information segments, facilitating subsequent retrieval of those segments in appropriate circumstances when their annotations match human-submitted queries.

In the course of the RKF effort, two significant accomplishments were attained. First, we developed a compact, graph-based representation for natural language that serves usefully as an interlingua between a natural language processing system such as START and external reasoning systems such as knowledge-based systems or knowledge acquisition systems. Second, we applied the graph-based representation by implementing an interface between the START system and the SHAKEN knowledge acquisition and reasoning system. This interface supports a suite of language-based capabilities.

The remainder of this report is divided as follows. Section 2 provides background material on the START information access system. Section 3 describes the graph-based representation of language and the "language-to-graph" translator we implemented as an extension of the START system. Section 4 describes the suite of language-based knowledge acquisition capabilities we implemented for START in conjunction with the SHAKEN system. Section 5 describes conclusions of the research. Finally, section 6 lists research publications generated during the course of our RKF effort, followed by a list of additional references cited within the report.

2. Background

Our START natural language question answering system has been in continuous development for nearly two decades (Katz, 1990; Katz, 1997). The use of natural language annotations, first introduced into START in the early '90s, enabled a quantum jump in question answering sophistication. Natural language annotations are short sentences and phrases associated with various information segments to help computers understand content they otherwise could not analyze. This metadata describes, in English, the types of questions that a particular piece of content can answer. Consider the following segment, which contains both text and an image:



The largest of the volcanoes in the Tharsis Montes region of Mars, as well as all known volcanoes in the solar system, is **Olympus Mons**. Olympus Mons is a shield volcano 624 km (374 mi) in diameter (approximately the same size as the state of Arizona), 25 km (16 mi) high…

The following annotations may be written to describe the segment:

> Picture of Olympus Mons
> Mars' highest point
> Largest volcano in the solar system
> Olympus Mons is 25km tall.

START parses these annotations and stores the parsed structures, called ternary expressions (Katz, 1988), with pointers back to the original information segment. To answer a question, the user query is analyzed and compared against the annotations stored in the knowledge base. If a match is found, the segment corresponding to the annotation is returned to the user as the answer. As an example, the annotations above would allow a question answering system to answer the following questions:

> Do you know what Olympus Mons looks like?
> What is the height of Olympus Mons?
> What is the highest point on Mars?
> Where can I find the largest volcano in the Solar System?

Because START matches questions against annotations at the level of syntactic structures, linguistically sophisticated machinery such as synonymy/hyponymy, ontologies, and structural transformation rules can be brought to bear on the matching process. These technologies allow our system to answer questions with far greater accuracy than can be achieved with traditional keyword-based systems. For example, a keyword-based system would incorrectly return the above segment to a user in response to the following unrelated questions:

> Are there volcanoes on the largest planet in the solar system?
> What is the largest volcano in Arizona?

Are all volcanoes in the Tharsis Montes region shield volcanoes?

The information segment about Olympus Mons shares many keywords with the questions listed above, yet it does not answer any of those questions. Instead of providing the wrong answer, START would be able to respond appropriately.

Natural language annotations can be attached to a variety of objects. Often, annotated segments may refer to a procedure in lieu of literal content. For example, the question "What time is it in Stockholm?" can match an annotation with a procedure that consults the computer's internal clock and time zone information. When the annotation triggers, START executes the relevant procedure; the result, typically couched in natural language, is then returned to the user.

The current START system is augmented by a supporting "virtual database" system called Omnibase (Katz *et al*., 2002). Omnibase provides uniform access to an open-ended, large variety of heterogeneous information sources—databases, web pages, textual documents and more through a stylized relational model that casts information in terms of *object-property-value* triples.

Natural language annotations can also be generalized by grouping words in annotations into classes of words. An annotation "parameterized" in this manner can match a set of related questions, for example, "When was *x* born?", where *x* can stand for any one of thousands of famous people. Our Omnibase system has a gazetteer function that identifies "symbols" along with their class names, allowing START to connect symbols in questions with their class names in annotations. This process, in conjunction with our Omnibase database technology, allows a single annotation to potentially answer hundreds of thousands of individual questions, e.g., "Who directed Titanic?", "Who directed Casablanca?", etc.

Natural language annotation technology and supporting capabilities allow the START system to create a large base of knowledge from diverse sources, yet organize it and standardize it in such a way as to make it usable by computers and humans. Traditional approaches to this difficult problem have fallen largely at two extremes: either attempt a wholesale encoding of knowledge in symbolic form (e.g., the Cyc project (Lenat *et al.*, 1990) or the ISI Sensus project (Knight and Luk, 1994)) or leave the knowledge in its original form—often multimedia information—and attempt to organize and index it for general use (e.g., the World Wide Web, HTML and XML). START draws benefit from both approaches by directly encoding knowledge where possible, yet relying on encoded annotations of knowledge in other cases.

On a deeper level, START takes the position of using simple natural language as a representation in its own right—that is, as an encoding from which reasoning and question answering can be performed. In this respect, START shares an intellectual heritage and viewpoint with work in semantic networks (e.g., Quillian, 1969), Concept Maps (e.g., Novak and Gowin, 1984; Novak, 1998) and restricted natural languages such as Ogden's Basic English (Ogden, 1968) and the ACE specification language (Fuchs *et*

al., 1999). START carries this ideology one step further, however, in providing an end-to-end question answering approach based on the idea of language as a representation—from free, natural language questions to standardized language questions, through matching to answers based on those questions, and finally to the generation of natural language and multi-media responses.

In 1993, START became the first natural language question answering system available on the World Wide Web. Since that time, START knowledge bases have been constructed to cover a number of domains, including: research and personnel at the MIT AI Laboratory, almanac-type information about cities and countries of the world, progress of the U.S. mission in Bosnia-Herzegovina, the Voyager spacecraft's encounter with the planet Neptune, military capabilities and interests of several Middle East countries and related organizations such as terrorist groups, information extracted from an introductory biology textbook, an ongoing START exhibit for the MIT Museum, and NASA-generated information and FAQ logs concerning the planet Mars. Since its introduction on the World Wide Web, START has answered millions of questions for hundreds of thousands of users around the world.

3. The Language-to-Graph Translator

3.1 Summary of Results

During the initial portion of the RKF program, MIT and the University of West Florida conducted a Component Experiment to test the integration of START with UWF's Concept Map Toolkit software for the purpose of interactively translating natural language text to a Concept Map representation of the text. In the course of the Component Experiment effort, we designed an interface language between the two software systems, implemented necessary changes to the systems to accommodate the interface, added specialized biology terminology to START's lexicon, designed an interactive process through which human and computer can collaboratively translate natural language text using the combined systems, and tested the setup on a set of randomly-selected text passages from the biology domain.

The results of our experiment were very encouraging. The interactive translation process was shown to be quite robust, with no deviations from the "flowchart" of prescribed activity required during the test examples. Of the 36 sentences tested, all were translated in substantially correct fashion—many yielding Concept Maps with 20 or more nodes and links—with a total of 5 minor errors arising, largely due to human oversight. Iterative refinement of input text during parsing—a key component of the interactive process—was reasonable in its occurrence, with each input sentence requiring on average the composition of 3 to 4 "subsentences" expressing the sentence's content in parsable ways, and with approximately one repeated pass through the parser required for every two subsentences translated.

Following is a brief summary of work accomplished in the course of the Component Experiment. In broad terms, the Component Experiment was divided into three stages of activity: design of the START/Concept Map interface and surrounding translation

process, implementation and refinement of the interface and process, and evaluation of the interface and process.

Design

Integration of the START system with the UWF's Concept Map software involved only a portion of each system's functionality. In particular, START's parser was employed, and the Concept Map rendering and editing functionality of the Concept Map Toolkit was employed. We chose a file-based transfer mechanism, whereby START's results of parsing were saved to an ASCII file for loading into the Concept Map Toolkit software.

START's internal representation is based on a form of concept-relationship-concept triples, as is the Concept Map software's graphical rendering of information. The key to interfacing the two systems was thus one of designing an intermediate representation consisting of concept-relationship-concept triples, but also including a specification of usage rules for concepts and for relationships. We designed such a representation by taking the following paragraph extracted from *Essential Cell Biology* (Alberts *et al.*, 1998), passing its sentences through the START parser, and hand-translating START's internal representations to a graphical form.

(*Essential Cell Biology*, Chapter 5, p. 155)

> The biological properties of a protein molecule depend on its physical interaction with other molecules. Thus, antibodies attach to viruses or bacteria as a signal to the body's defenses, the enzyme hexokinase binds glucose and ATP before catalyzing a reaction between them, actin molecules bind to each other to assemble into actin filaments, and so on. Indeed, all proteins stick, or *bind*, to other molecules. In some cases this binding is very tight; in others it is weak and short-lived. But the binding always shows great *specificity*, in the sense that each protein molecule can bind just one or a few molecules out of the many thousands of different molecules it encounters. Whether the substance that is bound by the protein is an ion, a small molecule, or a macromolecule, it is referred to as a *ligand* for that protein (from the Latin *ligare*, "to bind").

We then refined the representation through further experimentation, bringing it to its approximate, final form.

In parallel with this effort, we sketched the interactive human-computer translation process and decided on implementation particulars: which platforms and software packages to use for running the systems, how to maintain lists of sentences awaiting processing, and so forth.

Implementation

To implement the interface, we modified the START system to operate in a special mode under which it would accept assertions (rather than questions, as is normally the case for

START) and generate an output translation of each assertion as a list of node-link-node triples. The modified version of START was made to be accessible either through the World Wide Web or by running a modified Emacs session on an MIT Sun Workstation. An indexing mechanism was agreed upon whereby multiple occurrences of the same node in START's internal representation would be tagged with like numerical indices, so that the Concept Map software could detect these equivalences and avoid the generation of multiple nodes in its graphical rendering of START's output. The Concept Map software was likewise modified to accept an input file of node-link-node triples and translate these into a displayable Concept Map fragment.

Separately, START's lexicon of approximately 50,000 terms was augmented with about 2,000 specialized biology terms drawn from various sources, many within the *Essential Cell Biology* textbook. Following is a listing of these sources:

- the glossary of *Essential Cell Biology* (600 terms*)*
- the glossary of *Microbiology Webbed Out (Paustian, 2000)* (70 terms*)*
- examination of chapter-end material in *Essential Cell Biology* (100 terms*)*
- detailed analysis of the roughly 300 "Essential Concepts" listed in *Essential Cell Biology* (300 nouns and 90 adjectives)
- a word-frequency analysis of *Molecular Biology of the Cell, 3rd Edition* (Alberts *et al*., 1994) (100 terms*)*
- incorporation of a list of terms enumerated by the RKF project team at the University of Texas, drawn from an analysis of the text of *Molecular Biology of the Cell, 3rd Edition* (90 verbs*)*
- application of an MIT-developed colocation-detection tool to the text of *Molecular Biology of the Cell, 3rd Edition* (110 mostly multi-word terms*)*
- terms identified during implementation test runs of the combined systems (100 terms)
- additional nouns and adjectives from Chapters 1-7 of *Essential Cell Biology* (150 terms)
- terms utilized within a set of natural language annotations developed for the "Essential Concepts" in *Essential Cell Biology* (300 terms)

Next, in order to test and refine all aspects of the combined setup, we selected two paragraphs from each of the first five chapters of *Essential Cell Biology* and ran the sentences of these paragraphs through the combined systems. As a result of these processing runs, we made a number of modifications to the interface representation and START's lexicon, parsing rules and translation code. One such modification concerns the creation of new software that enables a human operator to inspect START's range of part-of-speech assignments applied to words and word sequences appearing in input sentences to the system.

Evaluation

Prior to evaluating the combined systems, we finalized our specification of the collaborative human-computer translation process and tested the process, its implementation and logging mechanisms on several test sentences. For the actual

evaluation runs, we randomly selected 7 paragraphs from *Essential Cell Biology*, one paragraph for each of the first 7 chapters. We then processed the sentences of the selected paragraphs using the collaborative human-computer translation process we had developed, operating on a PC and using the following software:

- START, accessed remotely through a (Secure Shell) SSH connection to an Emacs session running the START system on an MIT Sun workstation
- the Concept Map Toolkit running locally on the PC, with resultant Concept Maps saved as GIF files
- an Emacs session running locally on the PC, used both to maintain lists of sentences and subsentences awaiting processing, and to serve as a buffer for START's output (copied and pasted from the remote START Emacs window, then saved to a local ASCII file for loading into the Concept Map Toolkit)

The remote Emacs session was designed to log all aspects of START's processing and allow for the insertion of comments regarding the portion of the processing that was external to START. The log files were then processed for the loading of step-sequencing and timing information into an Excel spreadsheet, where various statistical relationships were calculated.

3.2 Human-Computer Collaborative Translation

This section describes the process used to translate natural language text into Concept Maps. The process can be portrayed as a flowchart with individual steps taken by either the human operator or the combined START and Concept Map Toolkit systems.

Abstractly, the translation process is composed of a few main steps. The human operator initially reformulates an input sentence as one or more "subsentences"—sentences that capture individual assertions made by the sentence. Also, some grammatical simplification can be performed by the human operator at this point. Next, START is asked to parse each subsentence in turn. If START fails to translate a subsentence, the human operator may rephrase the subsentence or decompose it further into additional subsentences, or the human operator may create or alter a START lexicon definition before retrying the subsentence. This process iterates until START succeeds in translating a subsentence, at which point the output representations from START are loaded into the Concept Map Toolkit and displayed as a Concept Map. As a final step, the human operator may optionally edit the Concept Map to correct any errors he or she detects.

The translation process flowchart is given below. The process makes use of two working lists: "sentence_list," which contains sentences awaiting processing, and "subsentence_list," which contains subsentences awaiting processing. Important step sequences are numbered for easy reference, and brackets are used to indicate portions of the process that are described more loosely in English.

```
    sentence_list = [all sentences to be processed]
    while [sentence_list is not empty] do
1       sub_sentence_list = [empty list]
```

```
1       s = pop_first(sentence_list)
1       [human operator optionally simplifies s syntactically while
         preserving its vocabulary, possibly generating multiple
         subsentences s1 .. sn]
1       [insert s, or s1 .. sn, as elements at the end of
         sub_sentence_list]
2       [clear the current Concept Map and reset START]
        while [sub_sentence_list is not empty] do
31          ss = pop_first(sub_sentence_list)
31          [human operator submits ss to START]
            if [START produces output triples] then
321             [human operator imports the output triples into the
                 Concept Map Toolkit software]
322             [human operator assesses the correctness of the
                 generated Concept Map]
                if [human operator finds the Concept Map to be
                    acceptable] then continue while
                else
3241                [human operator edits the resultant Concept Map]
                    end if
            else if [the failure is due to an undefined term] then
331             [human operator defines the term in question]
331             [push ss back onto the beginning of
                 sub_sentence_list]
            else if [the failure is due to a term that is already
                     defined] then
341             [human operator modifies the lexicon definition for
                 the term in question]
341             [push ss back onto the beginning of
                 sub_sentence_list]
            else if [the failure is due to parsing difficulties] then
351             [human operator modifies the syntax of ss or breaks
                 it into multiple subsentences ss1 .. ssn]
351             [push ss, or ss1 .. ssn, as elements onto the
                 beginning of sub_sentence_list]
                end if
            end while
        end while
```

Section 3.3 provides a detailed specification and examples of use for the interface language employed by START to convey its parsing results to the Concept Map Toolkit software. In summary, the representation consists of a list of node-link-node triples that conform to the following rules:

- Nodes are used to represent **nouns**, **verbs**, **adjectives** and **adverbs**
- Links are used to cover:
  - relationships between verbs and their arguments: **"has_subject", "has_object"**, plus **prepositions** for indirect objects
  - fundamental semantic relationships: **"is"** (for equality, membership, and subclass relationships), **"has"** (for possessives and related constructions)
  - modification of verbs: **"has_polarity"** (for negation), **"has_modifier"** (for adverbs), **"has_mode"** (for auxiliary verb sequences)
  - modification of nouns: **"has_quantifier", "has_quantity", "has_property"** (for adjectives)

- inter-event relationships and other relationships: **"has_method"** (accomplished by means of), **"has_purpose"** (having as a goal), and all **conjunctions** and **prepositions**

As an example, the input sentence

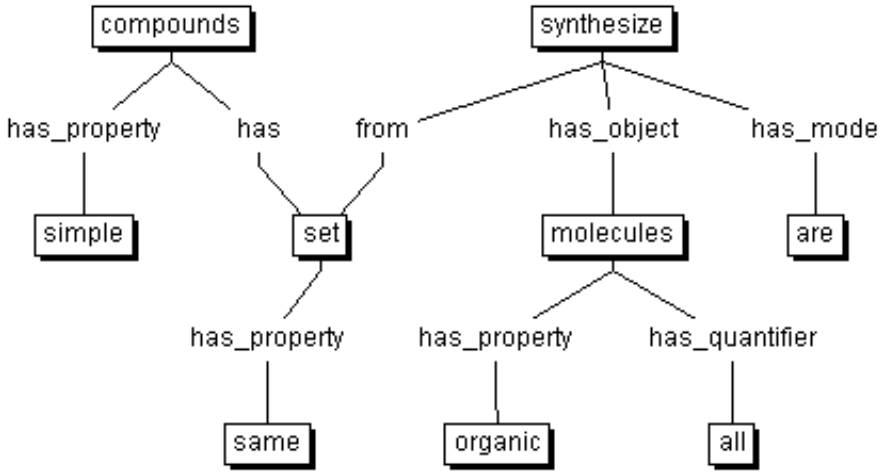
```
All organic molecules are synthesized from the same set of simple
compounds.
```

produces the following output when processed by START in step sequence 31, above:

```
[synthesize-1 has_object molecules-1]
[synthesize-1 from set-1]
[synthesize-1 has_mode are]
[set-1 has_property same]
[compounds-1 has set-1]
[compounds-1 has_property simple]
[molecules-1 has_quantifier all]
[molecules-1 has_property organic]
```

As it translates input text to output representations, START provides a standardized treatment of several aspects of the input, including: verb argument structure, possessive relationships, class-subclass and class-instance relationships, referring expressions, sentential embedding, negation, modifiers, and quantifiers. START also assigns index numbers to nodes in order to indicate multiple appearances of the same node, or, as the case may be, distinct nodes that would otherwise carry the same label.

If START successfully produces an output representation for an input sentence, the output triples are imported into the Concept Map Toolkit software in step 321, above, where they are displayed as a Concept Map. One Concept Map node is created for each distinct node name provided by START (minus the distinguishing index number), and links are drawn between the nodes according to the links specified in START's output triples. For the example listed above, the Concept Map fragment that is produced appears as follows:



If the human operator detects an error in the displayed Concept Map, he or she may edit the Concept Map in step 3241, above, by creating or eliminating nodes and links. In the case of the above Concept Map fragment, there are no errors to correct; however, in other cases there may be a parsing error by START—possibly resulting in the incorrect attachment of a particular link—or a reference error in which two nodes should be merged into one, or one node split into two. The human operator may also reposition nodes and links spatially if desired during step 3241.

If START does not successfully generate a list of output triples for an input subsentence, the human operator may optionally create or modify a START lexicon entry or rephrase the subsentence before resubmitting a subsentence to START. START lexicon entries contain a number of data fields; however, for the purposes of this translation process, it was found to be sufficient for the human operator to supply the following information in step 331 (creating a lexicon entry) or 341 (modifying a lexicon entry), depending on the part of speech:

- **nouns**: name, gender, proper noun status, mass noun status, irregular plural
- **verbs**: name, use in transitive and intransitive applications, irregular forms
- **adjectives**: name
- **adverbs**: name
- **prepositions**: name
- **conjunctions**: name

Rephrasing a sentence (step 351, also possible in step sequence 1) involves rewriting the sentence with altered grammatical structure or substitution of terms, or possibly breaking the sentence or subsentence into component subsentences, each of which conveys some portion of the meaning of the original sentence or subsentence. During the evaluation portion of this Component Experiment, rephrasings were typically carried out in such a way as to preserve as much as possible of the original wording of the sentence or subsentence.

3.3 A Graph-Based Representation of Language

This section details the interface language used by START to communicate its results of parsing to the Concept Map Toolkit software. The section begins with a description of START's input-output behavior when translating natural language assertions to lists of node-link-node triples.

The START RKF Server accepts individual assertions in simple English and generates one of four responses:

- A listing of symbolic triples constituting START's output to the Concept Map software, as detailed below.
- A message stating that a particular English term was not known to START, but that one or more close matches were found in the lexicon. The user may optionally select one of these alternatives in order to proceed with the current parse. (A close match is one that omits or adds a single letter or transposes two letters of a word.)
- A message stating that a particular English term was not known to START, yet offering no close matches. The user must resubmit the sentence.
- A message stating that START could not parse the input and asking the user to rephrase the input.

If a list of triples is returned, the list appears as a contiguous sequence of triples separated by carriage returns. Each triple corresponds to a node-link-node relationship in the generated Concept Map and consists of a left bracket character ("["), followed by three symbolic tokens separated by spaces, followed by a right bracket character ("]"). Elsewhere in the response web page or ASCII output there may be sequences that contain bracket characters, but no sequences in which brackets surround three symbolic tokens separated by spaces. Each symbolic token consists of a sequence of uppercase and lowercase letters, digits, hyphens ("-") and underscore characters ("_").

In the first and third token positions of triples, tokens may be given a suffix consisting of a hyphen followed by a non-negative integer. These suffixes are used to indicate equality between generated Concept Map nodes (e.g., "cell-3" in one triple is taken to refer to the same node as "cell-3" in another triple, and "cell-5" is taken to refer to a different node in the Concept Map). No tokens containing a suffix sequence will be generated unless it is intended for the suffix to be used as an index in this manner. Also, the suffix construction is optional: some symbolic tokens in the first and third token positions will not contain suffixes. In addition, all tokens in the second token position (the Concept Map "link") will not carry such suffixes. Elsewhere within each symbolic token, an underscore character is used whenever a space would separate the words of a multi-word English term (e.g., "immune system" is cast as "immune_system"), and a hyphen is used whenever a hyphen would separate the words of a multi-word English term (e.g., "short-lived" or "membrane-bounded").

To generate the output triples, START analyzes the input sentence to identify English terms, their parts of speech, and their grammatical relationships within the sentence. All nouns, verbs, adjectives and adverbs are cast as Concept Map nodes, and prepositions and

conjunctions, along with a number of pre-defined relationships, are cast as Concept Map links. Thus, the Concept Map nodes come from a largely open set, and the links come from a largely closed set. For the Concept Map links, the possibilities are listed below, grouped into five categories. Examples illustrate the use of each link type.

Links between verbs and their arguments

has_subject - subject of an active voice rendering of the sentence
has_object - object of an active voice rendering of the sentence
(indirect objects are cast using prepositions "to", "for", "of", etc.)

Example:

```
"The plasma membrane gives the cell protection."

==>  [give-1 has_subject plasma_membrane-1]
     [give-1 has_object protection-1]
     [give-1 to cell-1]
```

Fundamental semantic relationships

is - for equality, membership and subclass relationships
has - for possessives and related constructions

Examples:

```
"Hydrogen is the lightest element."

==>  [is-1 has_subject hydrogen-1]
     [is-1 has_object element-1]
     [hydrogen-1 is element-1]
     [element-1 has_property lightest]

"The material is a substrate."

==>  [is-1 has_subject material-1]
     [is-1 has_object substrate-1]
     [material-1 is substrate-1]

"Enzymes are proteins."

==>  [is-1 has_subject enzymes-1]
     [is-1 has_object proteins-1]
     [enzymes-1 is proteins-1]

"The ligand regulates the protein's activity."

==>  [regulate-1 has_subject ligand-1]
     [regulate-1 has_object activity-1]
     [protein-1 has activity-1]
```

```
"The interior of the cell has a nucleus."

==>  [have-1 has_subject interior-1]
     [have-1 has_object nucleus-1]
     [interior-1 has nucleus-1]
     [cell-1 has interior-1]
```

Modification of verbs

has_polarity - for negation
has_modifier - for adverbs
has_mode - for auxiliary verb sequences

Example:

```
"Typically, the mechanism should not make mistakes."

==>  [make-1 has_subject mechanism-1]
     [make-1 has_object mistakes-1]
     [make-1 has_modifier typically]
     [make-1 has_mode should]
     [make-1 has_polarity not]
```

Modification of nouns

has_quantifier
has_quantity
has_property - for adjectives

Example:

```
"Each enzyme catalyzes one specific reaction."

==>  [catalyze-1 has_subject enzyme-1]
     [catalyze-1 has_object reaction-1]
     [reaction-1 has_property specific]
     [reaction-1 has_quantity 1]
     [enzyme-1 has_quantifier each]
```

Other link types

has_method - one event is accomplished by means of another
has_purpose - one event has another as its purpose
(conjunctions)
(prepositions)

Examples:

```
"The ligand operates by binding to the ion channel."

==>  [operate-1 has_method bind-1]
     [operate-1 has_subject ligand-1]
     [bind-1 has_subject ligand-1]
     [bind-1 to ion_channel-1]

"Cells use enzymes to catalyze chemical reactions."

==>  [use-1 has_purpose catalyze-1]
     [use-1 has_subject cells-1]
     [use-1 has_object enzymes-1]
     [catalyze-1 has_subject cells-1]
     [catalyze-1 has_object chemical_reactions-1]

"The free energy is captured when a fuel molecule is
oxidized in a cell."

==>  [capture-1 when oxidize-1]
     [capture-1 has_object free_energy-1]
     [oxidize-1 has_object fuel_molecule-1]
     [oxidize-1 in cell-1]
```

4. Language-Based Capabilities for Knowledge Acquisition

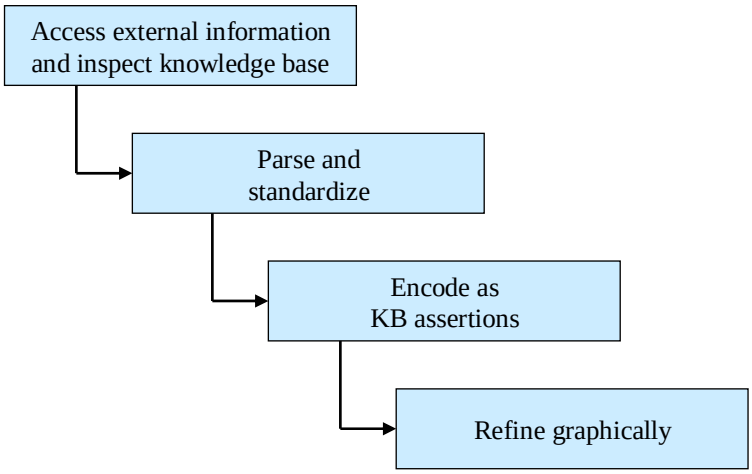
During the latter portion of the RKF program, we applied our language-to-graph translator to the problem of knowledge acquisition. In this part of the effort, we constructed an interface between START (including the language-to-graph translator) and the SHAKEN knowledge acquisition and reasoning system (Barker *et al*., 2003). This section describes the integration effort and functionality created.

4.1 Staged, Interactive Knowledge Acquisition

At the onset of this research effort, we hypothesized that an effective way to bring automated language processing into use during knowledge acquisition would be to break up the knowledge acquisition process into a sequence of stages, interspersed with system-human interaction. We expected two benefits of this approach:

- The high degree of interaction can help bring human language processing capabilities and human judgment to bear in the knowledge acquisition process, and
- Breaking the knowledge acquisition process into stages can help isolate functional units of the approach, which can aid in testing, debugging and modification.

Our initial hypothesis has been validated during the course of the research effort, and we have experienced both expected benefits as well. Following is a graphical characterization of the major stages of processing in our integrated, language-based knowledge acquisition system.



The first stage involves identifying the particular information to be entered and comparing that information to current knowledge base contents. The second stage concerns translation of an original text form of the information to a human-readable representation that is standardized in both form and content. This second stage is accomplished by our language-to-graph translator. The third stage involves translation of the standardized representation into assertions to be entered into the knowledge base. The fourth step involves human inspection and graphical refinement of the encoded assertions. Each stage may be further decomposed into substages. For example, the second stage, parsing and standardization, can be decomposed into recognition of terms in the input statement, analysis of the syntactic structure of the statement, transformation of terms to standardized terms, and transformation of syntactic structures to standardized syntactic structures.

In all, the staged translation of natural language statements to knowledge base assertions yielded numerous opportunities for system-human interaction. Following is a list of these interactions:

- defining new English terms
- rephrasing input sentences
- mapping English terms to knowledge base concepts
- refining knowledge base encodings graphically
- asking English questions
- matching phrases and sentences to knowledge base concepts
- attaching annotations to knowledge base concepts

4.2 A Suite of Language-Based Capabilities

Language processing can contribute in several ways to the knowledge acquisition process. In this research effort, we focused on three complementary capabilities:

- **knowledge retrieval**, in which the user inspects information already contained in the knowledge base or available from external resources,
- **knowledge organization**, in which the user catalogs knowledge base contents and external information for expedited future retrieval, and
- **knowledge entry**, in which the user translates externally obtained information into assertions which are entered into the knowledge base.

These capabilities can be combined in a number of ways during the knowledge acquisition process. For example, a user may retrieve existing knowledge base information, observe a missing component, retrieve external information, enter the external information into the knowledge base, add notations to mark the external resource for future use, and add notations to the added knowledge for future retrieval when related knowledge is added.

In all, we implemented five distinct capabilities for the SHAKEN system, using START and our language-to-graph translator. These are:

1. **Annotation-based knowledge retrieval,** which employs matching of questions to natural language annotations as a means of identifying answers.
2. **Relay-based knowledge retrieval,** which translates questions to resource-specific queries to be processed by other RKF systems.
3. **Match-based knowledge retrieval,** which translates natural language statements into knowledge patterns that are then matched to specific knowledge base entries.
4. **Annotation-based knowledge organization,** which allows users to compose natural language phrases and sentences that serve to describe the content of information segments the user wishes to make retrievable by the START system.
5. **Translation-based knowledge entry,** which transforms natural language statements into system-specific knowledge structures by performing a series of partial transformations that culminate in the execution of knowledge-structure creation code.

For all five techniques, we additionally focused on designs that would lead to domain portability and robustness. Domain portability of the techniques was indeed tested as we converted the techniques from an initial application in the domain of biology textbook material to a subsequent application in the domain of Intelligence Preparation of the Battlefield (IPB). In the course of this conversion, we identified four steps that must be taken in order to port the techniques to a new domain:

- We must insert new domain vocabulary into START's lexicon.
- We must insert SHAKEN concept names from the new domain into START's lexicon.
- We must attach natural language annotations to a core set of background documents and resources for the new domain.
- We must incorporate new SHAKEN question types into our Relay-Based Knowledge Retrieval capability.

Regarding robustness of the techniques, several features of the implemented, integrated system help achieve this goal:

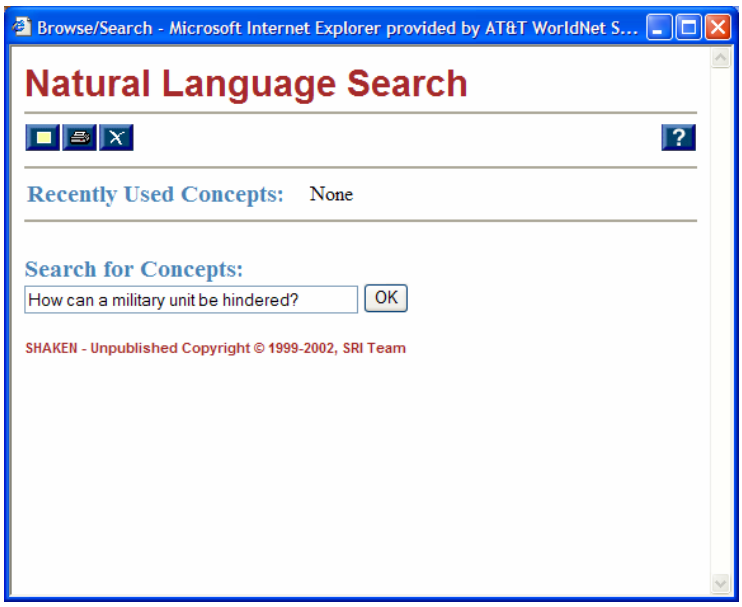
- The capabilities complement one another and create synergies.
- The user may interleave use of the five language-based techniques with conventional SHAKEN processing operations.
- All of the language-based capabilities are steerable through user interaction.
- In several cases, backtracking and iteration are used to facilitate successive refinement.
- The capabilities include error handling for unknown words, misspelled words and unparsable syntax.
- The knowledge retrieval capabilities can be augmented to apply to general resources outside the SHAKEN system.
- START's operation is user-extensible, allowing users to add new lexicon entries and add new natural language annotations.

The next five subsections describe each of the implemented capabilities in greater detail.

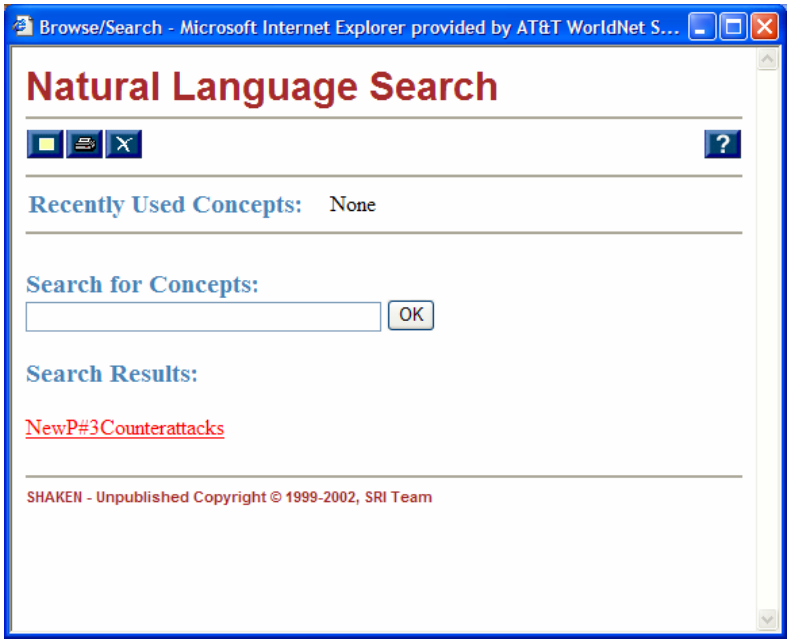
4.2.1 Annotation-Based Knowledge Retrieval

This technique uses START's natural language annotation strategy to answer questions submitted by the user, possibly retrieving information from the SHAKEN knowledge base or external resources. In the operation of this technique, the SHAKEN user submits a natural language question, which is forwarded to the START system. If START is able to respond with the names of one or more SHAKEN concepts, then these concepts are displayed in the original SHAKEN window. Otherwise, a START dialog window appears, through which the user may inspect other answers returned by START, rephrase the query, correct misspelled words, and define new words.

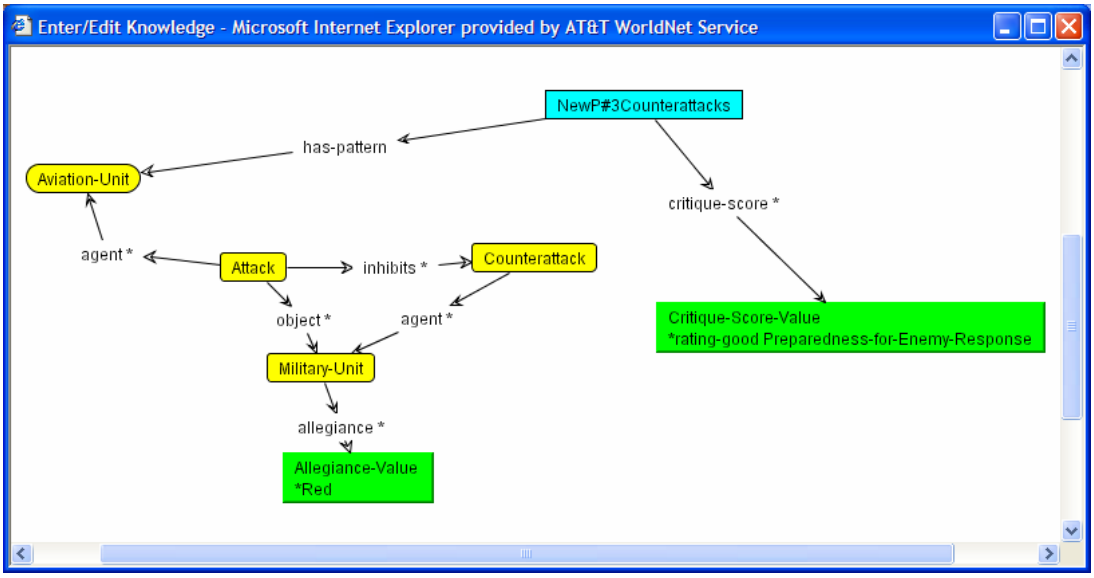
The following is an example of the use of Annotation-Based Knowledge Retrieval to locate a SHAKEN concept. Here, the user of a particular Intelligence Preparation of the Battlefield (IPB) knowledge base has entered a question "How can a military unit be hindered?":



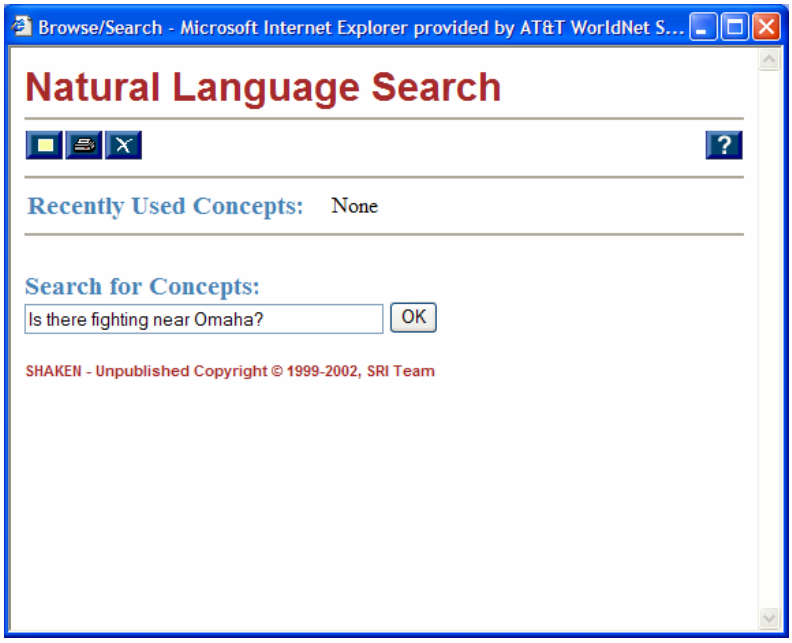
START matches the question to an annotation attached to a SHAKEN concept and returns the SHAKEN concept:



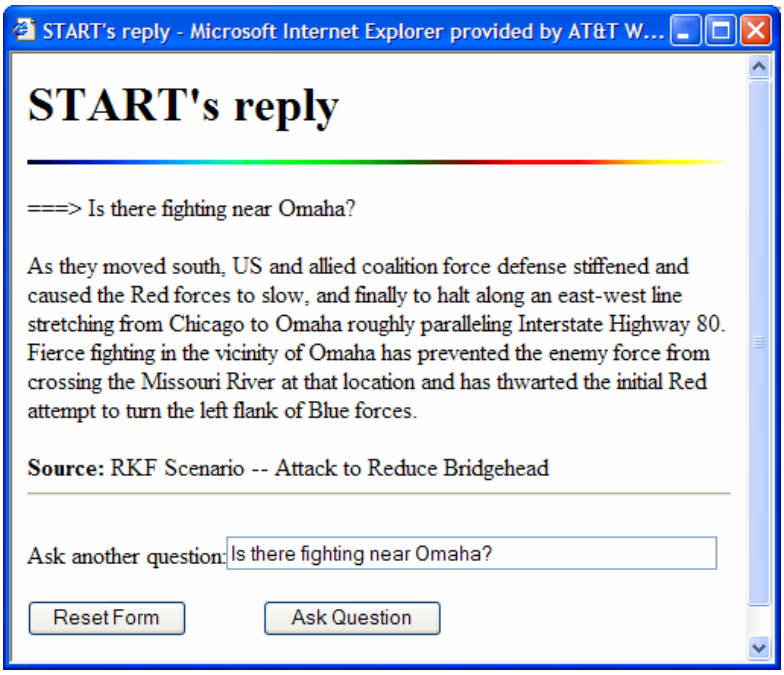
Next, the user may select the concept to view its definition within SHAKEN:



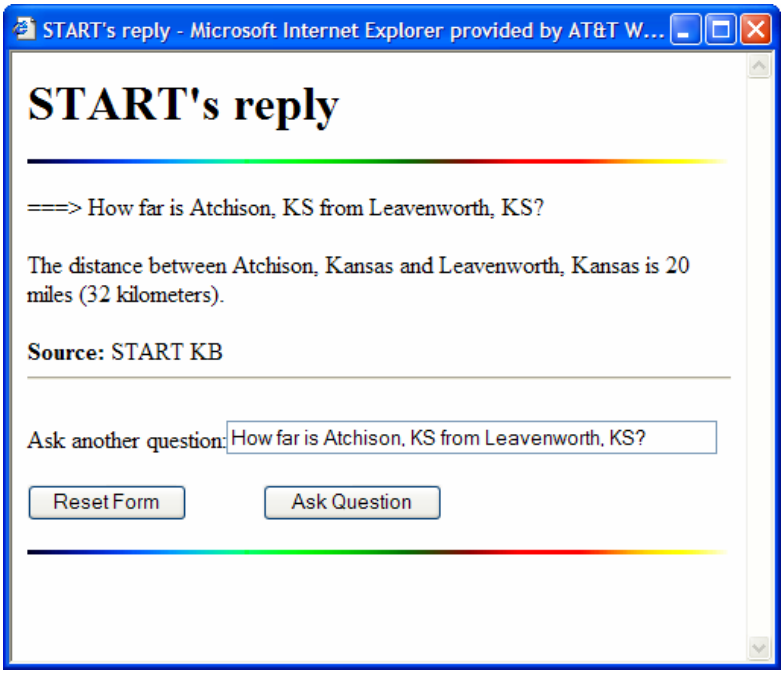
In a similar manner, the user may also retrieve background information related to the knowledge acquisition task. In the following example, the user submits a question about the circumstances depicted in the Attack to Reduce Bridgehead scenario.



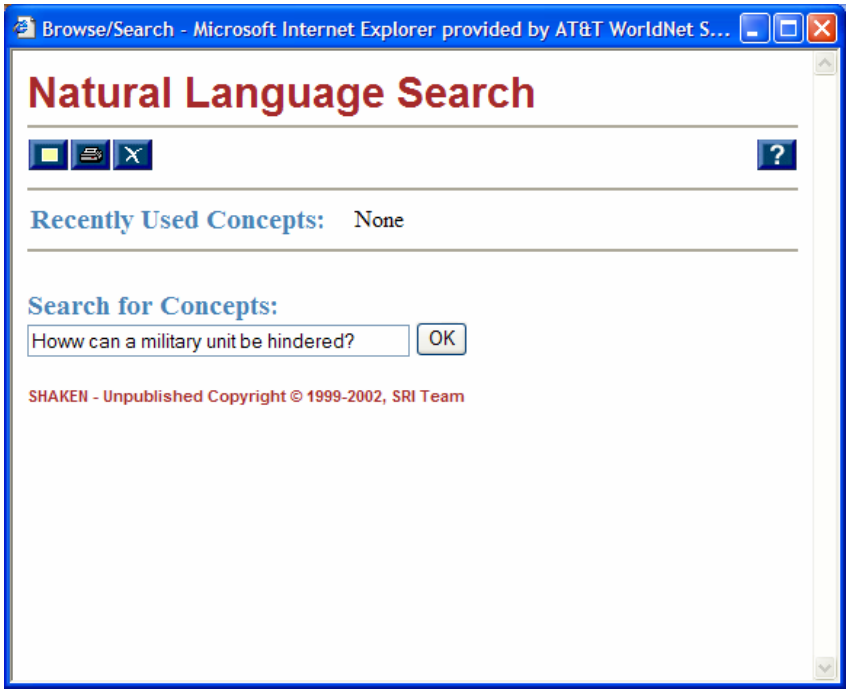
Since the answer is not a SHAKEN concept, START opens a new window to display the answer to the user:



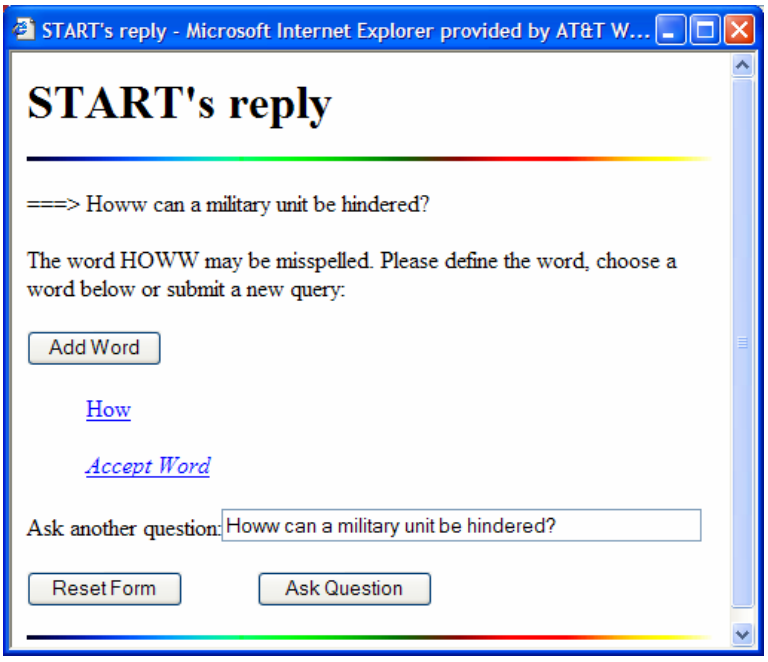
In the following example, START is used to retrieve general-purpose information from an external resource. Here, the user has asked for the distance between Atchison and Leavenworth, Kansas:



The next example illustrates a situation in which an extended dialog is required. START opens a new window for this purpose as well. Here, the user misspells the word “How” in the submitted question:

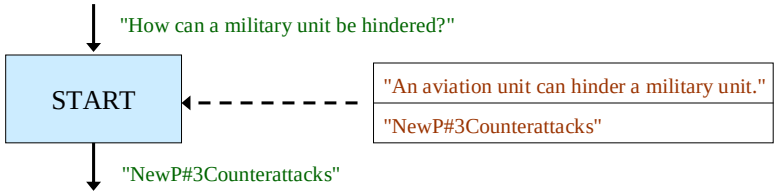


START responds with a dialog window that suggests a possible correction of the entered word:



Once the user selects the correct spelling “How”, the processing continues as in the first example.

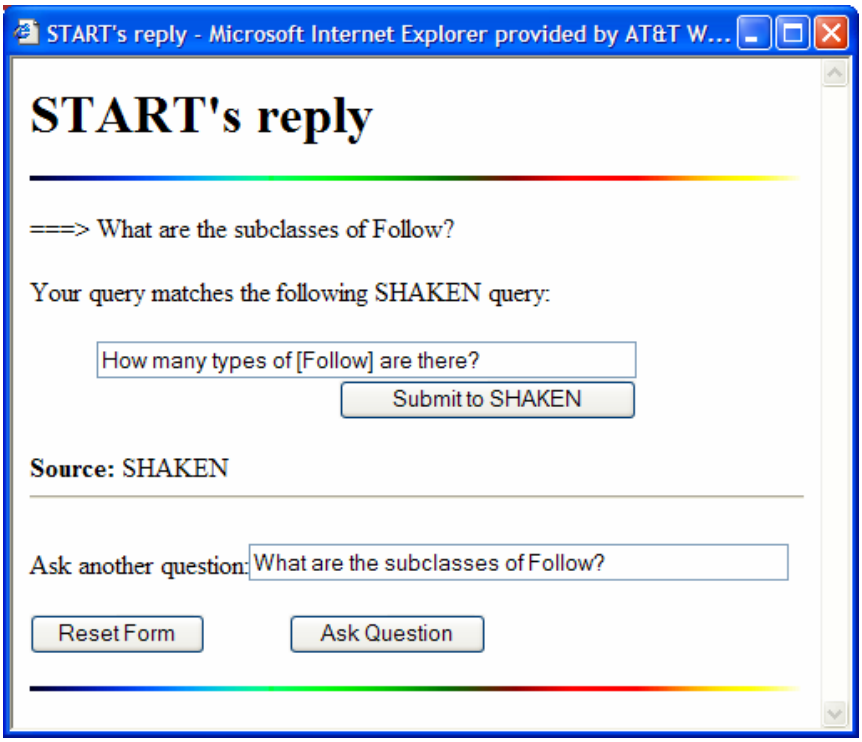
The following diagram shows how the Annotation-Based Knowledge Retrieval capability works. In this diagram, green is used to depict inputs and outputs, and brown is used to illustrate functionality. In the illustrated instance, the annotation "An aviation unit can hinder a military unit." has previously been associated with the SHAKEN concept "NewP#3Counterattacks". The user's question "How can a military unit be hindered?" is then matched to the stored annotation, and the attached concept is returned to the user.



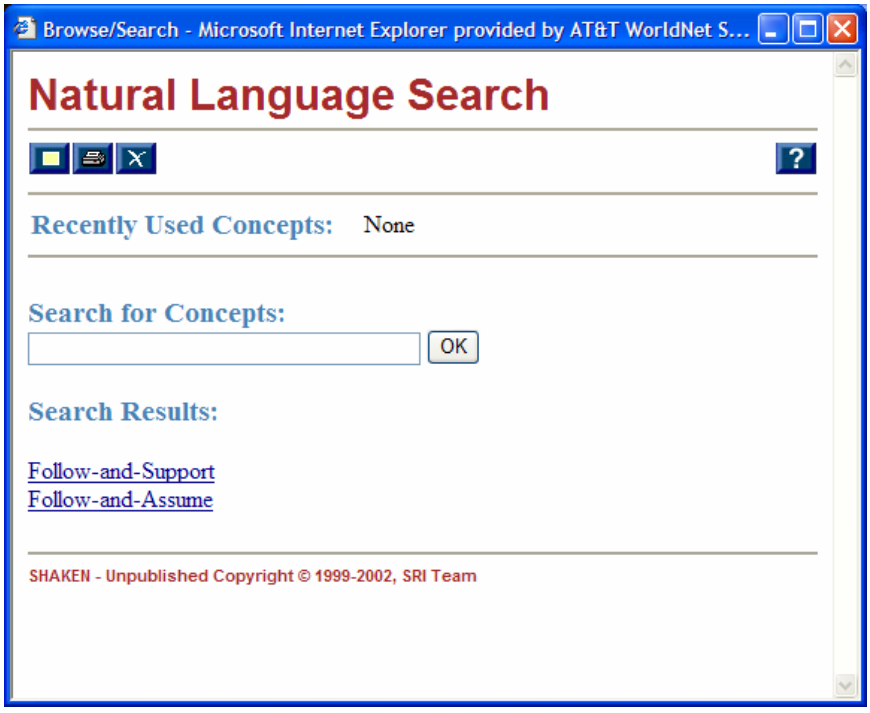
4.2.2 Relay-Based Knowledge Retrieval

Relay-Based Knowledge Retrieval works in a similar manner to Annotation-Based Knowledge Retrieval, except that the retrieved information is not a SHAKEN concept, but a query that can be forwarded to SHAKEN to obtain an answer. This takes advantage of SHAKEN's built-in functions for answering certain types of questions and makes these question types accessible through natural language.

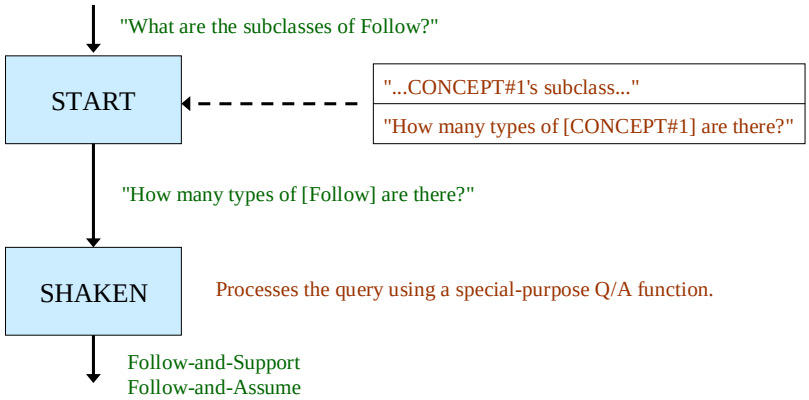
In the example that follows, the user has asked for subclasses of the SHAKEN concept "Follow". START opens a dialog window to request confirmation that this query may be passed on to SHAKEN:



When the user selects “Submit to SHAKEN”, the query is processed, resulting in the display of SHAKEN’s returned results to the user:



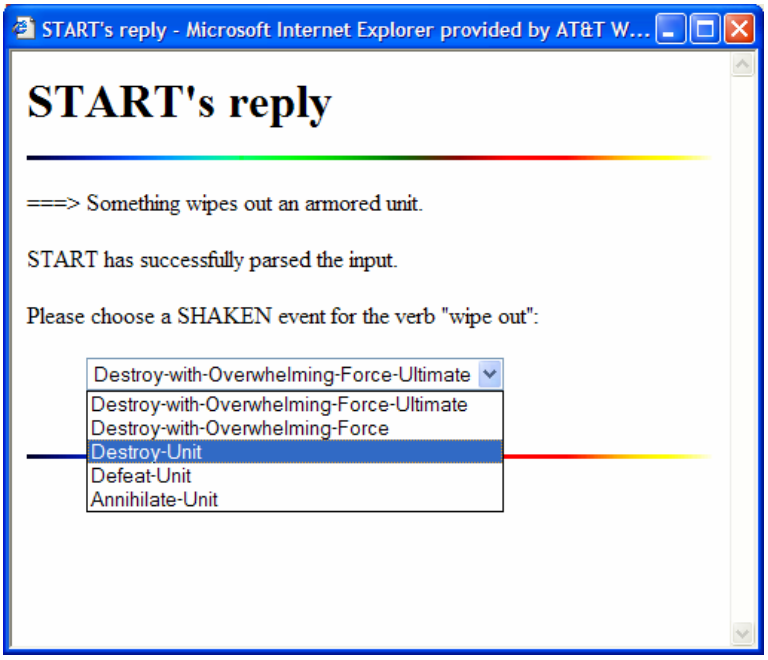
Relay-Based Knowledge Retrieval makes use of START’s annotation-based matching functionality to match an incoming question to an annotation containing one or more pattern variables. The annotated object is a pattern for a query to SHAKEN, and bindings formed during the matching of question to annotation are used to fully instantiate the SHAKEN query pattern, so that it may be submitted to SHAKEN. In the diagram below, the matching variable is “CONCEPT#1”.



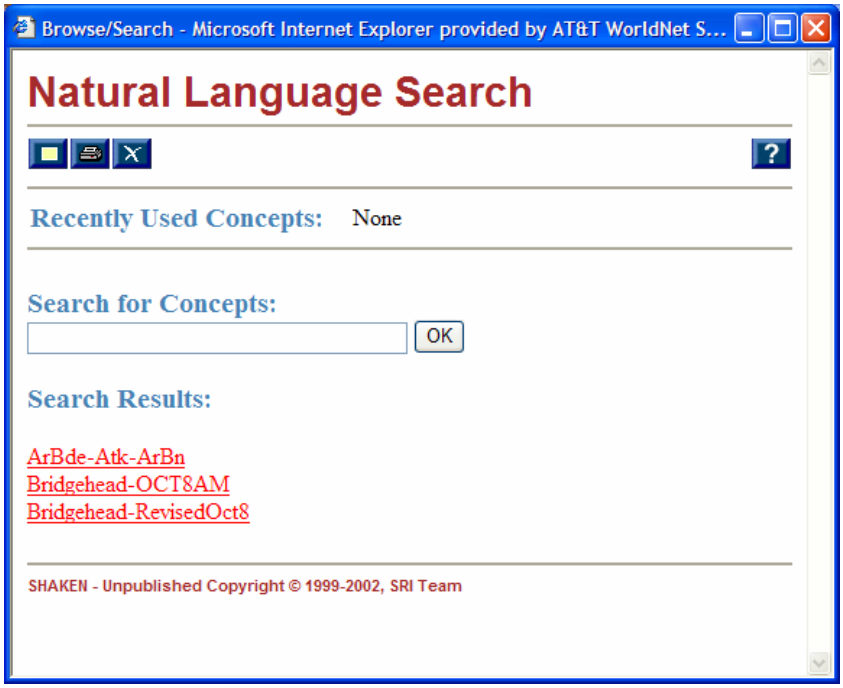
4.2.3 Match-Based Knowledge Retrieval

For Match-Based Knowledge Retrieval, the user enters either a sentence or phrase, and the system searches for SHAKEN concepts that contain that knowledge fragment within their defined Concept Maps. START processes the input using its language-to-graph translator, then incrementally and interactively maps each term to a corresponding SHAKEN counterpart. Finally, START's parsed input plus the term mappings are sent to a matcher that scans SHAKEN's knowledge base for matches.

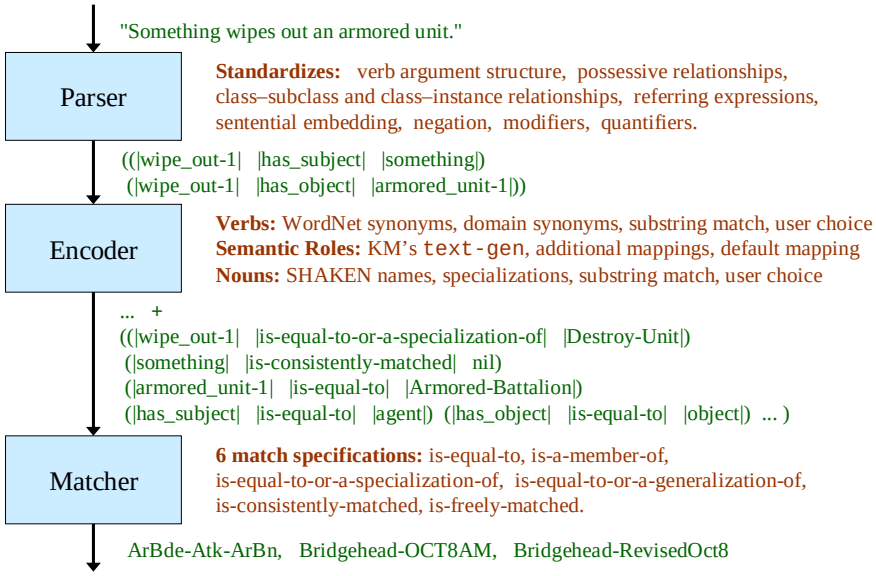
In the example that follows, the user enters a sentence "Something wipes out an armored unit." START applies its language-to-graph translator, then identifies candidate SHAKEN concepts for the verb and its arguments. Here, the user selects the SHAKEN concept "Destroy-Unit" for the verb "wipe out":



In a similar manner, the user selects "Armored-Battalion" for "armored unit", and three matches are identified in SHAKEN's knowledge base:



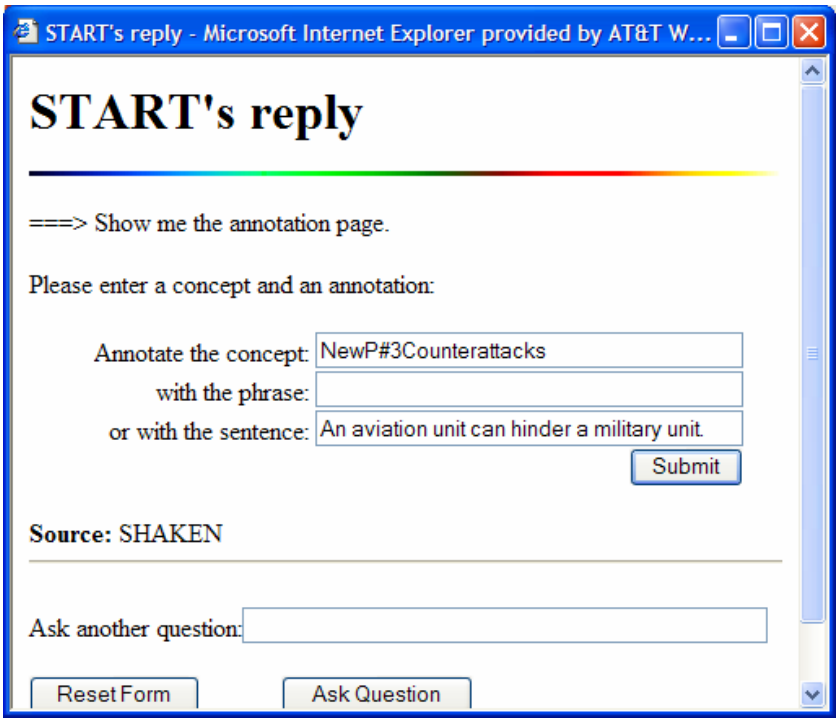
Match-Based Knowledge Retrieval is implemented as illustrated in the diagram below. START's language-to-graph translator standardizes several aspects of the natural language input and produces a set of node-link-node triples as output. The encoder then interactively composes a match specification for each term in the triples. Verbs, semantic roles and nouns are handled in slightly different ways. The triples plus the match specifications are sent to a knowledge base matcher that has been inserted within SHAKEN. The matcher handles six match specifications, listed in decreasing order of how much they constrain the values that may be accepted as matches for a term in the input triples. Finally, matched concepts are returned to the user.



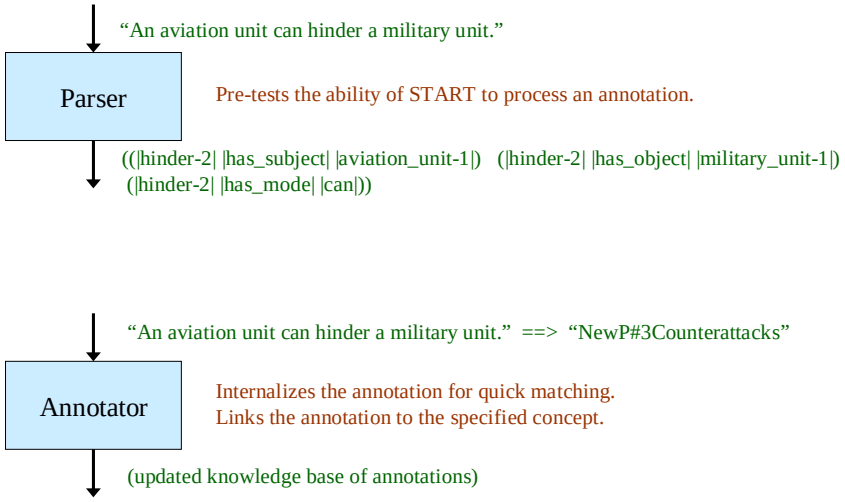
4.2.4 Annotation-Based Knowledge Organization

Using the Annotation-Based Knowledge Organization capability, the user may designate a SHAKEN concept to be augmented with natural language annotations and then supply phrases and sentences to act as those annotations. START will test the phrases and sentences for parsability, then associate them with the designated concept. The user may then access the concept through the use of natural language questions processed by the annotation-based retrieval technique.

In the following example, the user has selected the concept "NewP#3Counterattacks" and added a sentence annotation "An aviation unit can hinder a military unit."



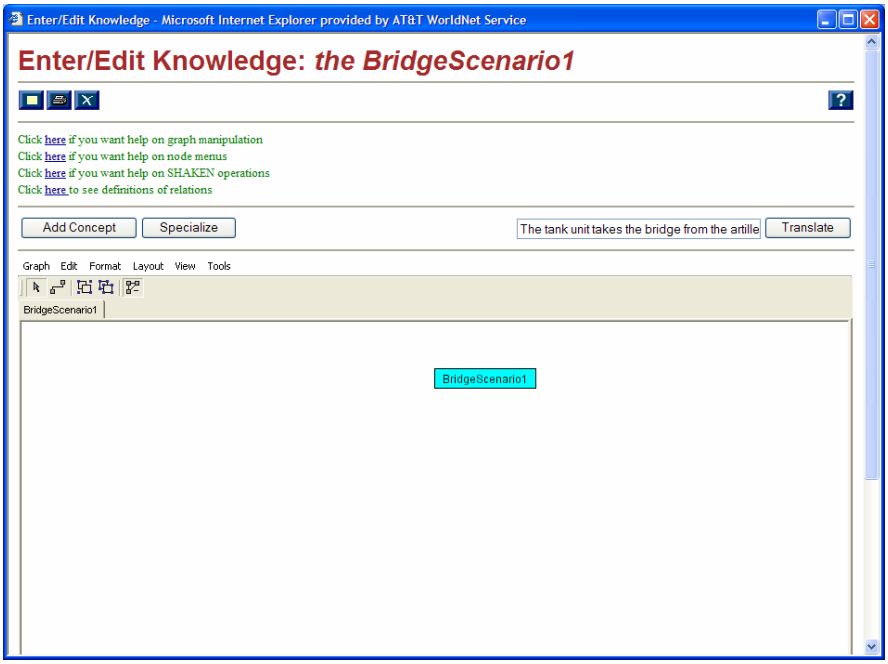
The implementation of this capability is illustrated in the diagram below. START's parser is used to pre-test annotations for acceptability. The user is provided with an opportunity to correct unacceptable annotations. The Annotator then submits the annotation entries into START's base of annotations.



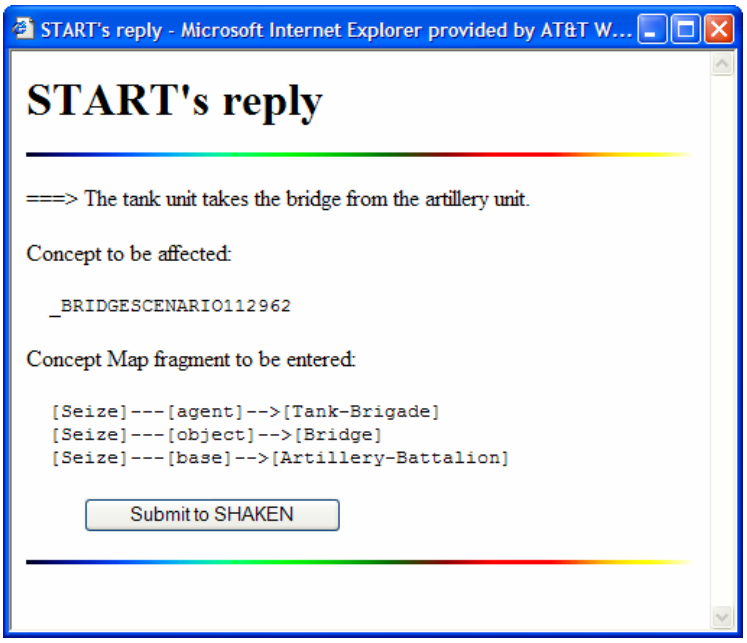
4.2.5 Translation-Based Knowledge Entry

Translation-Based Knowledge Entry is similar to the Match-Based Knowledge Retrieval, except that instead of creating a Concept Map pattern to be used for matching, START creates a fully-instantiated Concept Map fragment to be entered within the SHAKEN Concept Map for a designated SHAKEN concept.

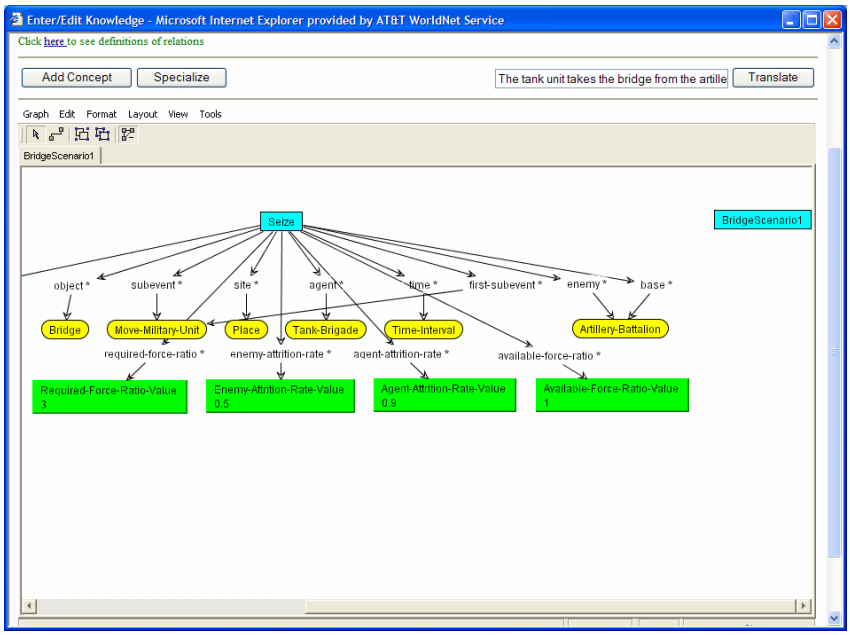
In the following example, the user creates a new SHAKEN concept "BridgeScenario1", then enters the sentence "The tank unit takes the bridge from the artillery unit."



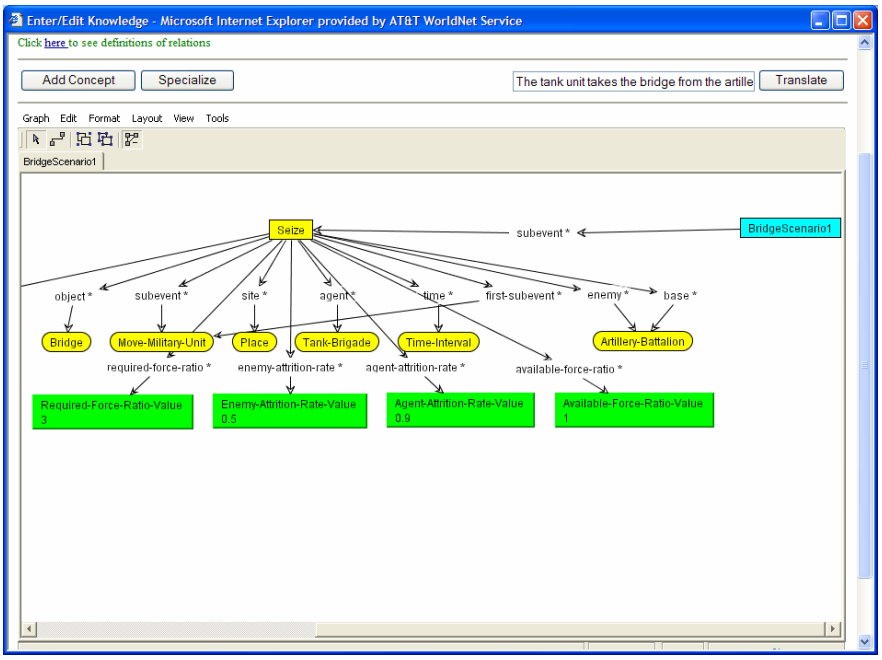
START processes the input sentence with its language-to-graph translator, then identifies candidate SHAKEN concepts to replace the verb and its arguments. The user selects specific SHAKEN concepts to use, and START completes the translation:



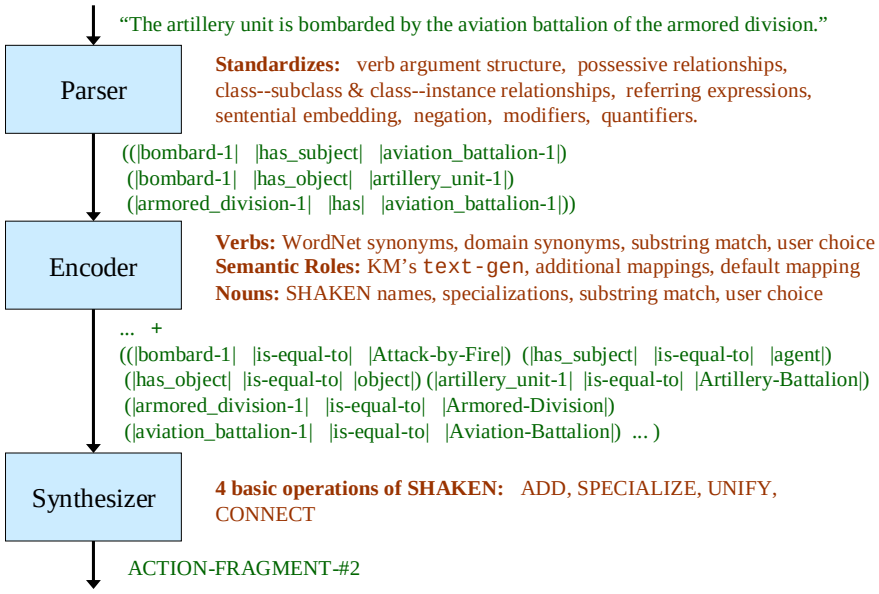
After the user has accepted the translated fragment, START adds the nodes and links to the target concept and returns it to the user. The user may then inspect the concept, edit the nodes and links graphically, add other nodes and links using SHAKEN's standard graphical operations, or add further nodes and links using the Translation-Based Knowledge Entry technique. In this example, the user first inspects the nodes and links, which have been added as a disconnected subgraph in the Concept Map for "BridgeScenario1".



The user then uses a graphical SHAKEN operation to connect the new fragment to the node “BridgeScenario1” using a “subevent” link.



The implementation of the Translation-Based Knowledge Entry technique is similar to that of the Match-Based Knowledge Retrieval technique, except that the matcher at the end is replaced by a knowledge “synthesizer” that uses native SHAKEN operations (ADD, SPECIALIZE, UNIFY and CONNECT) to create nodes and links within SHAKEN Concept Maps.



5. Conclusions

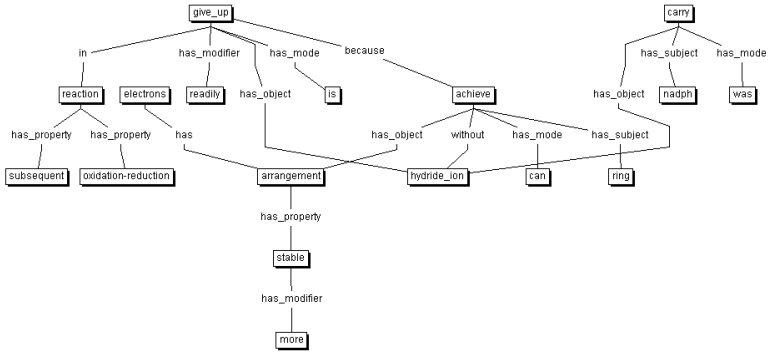
This research has supplied two main contributions. First, it has demonstrated the utility of staged, interactive translation of language to knowledge for purposes of knowledge acquisition. Second, it has elaborated a detailed specification for an exportable, labeled graph representation for parsed natural language for use in knowledge entry, knowledge matching, and mixed-initiative parsing.

If we contrast the two major implementation efforts described in sections 3 and 4—the language-to-graph translator and the suite of language-based capabilities for knowledge acquisition—we see that we have addressed two slightly different parts of the larger problem of language-facilitated knowledge acquisition. The language-to-graph translator accepts reasonably complex input sentences—from a biology textbook—and casts them into a relatively shallow semantic representation, in which English terms are preserved as nodes within graph-based representations. In contrast, the suite of language-based capabilities for START and SHAKEN accepts somewhat simpler English inputs, yet maps them all the way into knowledge base assertions. The next step, then, would be to extract insights from both efforts in an attempt to translate more complex input sentences into deeper semantic representations.

The following example illustrates the operation of the language-to-graph translator. Given a fairly complex input sentence:

> "The hydride ion carried by NADPH is given up readily in a subsequent oxidation-reduction reaction, because the ring can achieve a more stable arrangement of electrons without it."

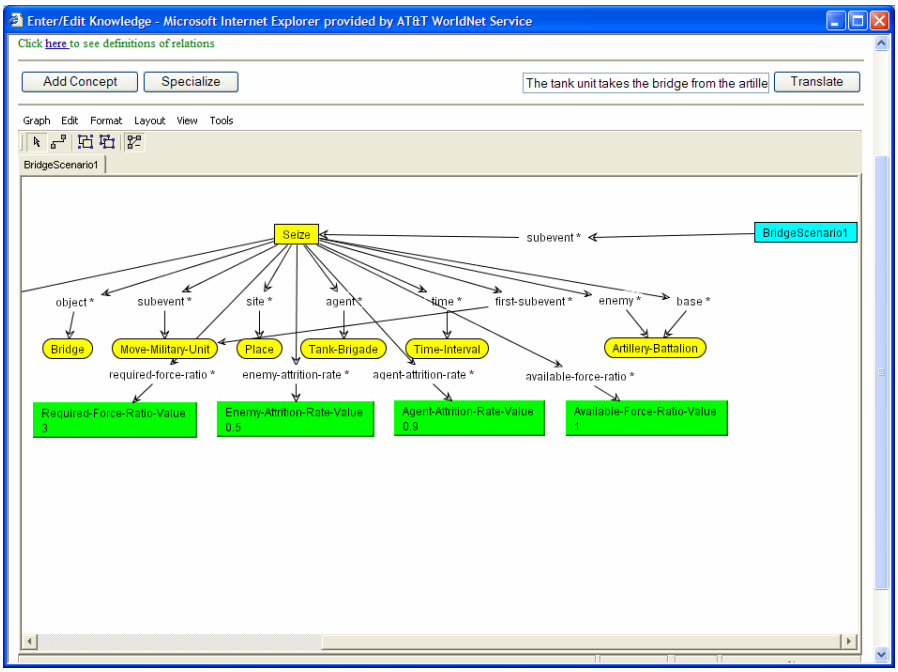
The translator produces the following graph:



By interactively processing knowledge fragments in their original form—free text—the language-to-graph translator takes a measure of burden away from the human user in the initial stages of the knowledge acquisition process.

On the other hand, when the Translation-Based Knowledge Entry capability of the START-SHAKEN integrated knowledge acquisition system is used, the system accepts a

simpler sentence such as “The tank unit takes the bridge from the artillery unit.” and generates a detailed knowledge encoding of that fragment.



This takes a measure of burden away from the human user in the latter stages of the knowledge acquisition process.

To fill the gap between these two techniques, we intend to explore three promising approaches. First, we’d like to take advantage of technology we’ve developed recently to automatically extract semantic relations from free text, even when the text itself is beyond the parsing capabilities of current parsers. Second, it may be possible to exploit human interactivity to reconcile the sorts of graphs produced by our language-to-graph translator and the sorts of graphs produced by knowledge entry within SHAKEN. We hope that this might even be attainable by individuals other than knowledge engineers or subject matter experts. Finally, we hope to exploit the complementary nature of knowledge retrieval, knowledge organization and knowledge entry techniques to assist one another in the overall knowledge acquisition process.

6. Publications

The following articles were published during the course of the START Rapid Knowledge Formation effort.

Katz, B., Lin, J. and Felshin, S. “Gathering Knowledge for a Question Answering System from Heterogeneous Information Sources,” in *Proceedings of the ACL 39th Annual Meeting, 10th Conference of the European Chapter, Workshop on Human Language Technology and Knowledge Management*, Toulouse, France, July 2001.

Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland, A. J. and Temelkuran, B. "Omnibase: Uniform Access to Heterogeneous Data for Question Answering," in *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002)*, June 2002.

Katz, B. and Lin, J. "START and Beyond," in *Proceedings of 6th World Multiconference on Systemics, Cybernetics, and Informatics (SCI 2002)*, July 2002.

Katz, B. and Lin, J. "Annotating the Semantic Web Using Natural Language," in *Proceedings of the 2nd Workshop on NLP and XML (NLPXML-2002) at COLING 2002*, September 2002.

Katz, B., Lin, J. and Felshin, S., "The START Multimedia Information System: Current Technology and Future Directions," in *Proceedings of the International Workshop on Multimedia Information Systems (MIS 2002)*, October 2002.

Katz, B., Lin, J. and Quan, D. "Natural Language Annotations for the Semantic Web," in *Proceedings of the International Conference on Ontologies, Databases, and Application of Semantics (ODBASE 2002)*, October 2002.

Lin, J., Fernandes, A., Katz, B., Marton, G., and Tellex, S., "Extracting Answers from the Web Using Data Annotation and Data Mining Techniques," in *Proceedings of the 2002 Text REtrieval Conference (TREC 2002)*, November, 2002.

Karger, D., Katz, B., Lin, J., and Quan, D., "Sticky Notes for the Semantic Web," in *Proceedings of the 2003 International Conference on Intelligent User Interfaces (IUI 2003)*, January, 2003.

Katz, B., Lin, J., Stauffer, C., and Grimson, E., "Answering Questions about Moving Objects in Surveillance Videos," in *Proceedings of 2003 AAAI Spring Symposium on New Directions in Question Answering*, March, 2003.

Katz, B. and Lin, J., "Selectively Using Relations to Improve Precision in Question Answering," in *Proceedings of the EACL-2003 Workshop on Natural Language Processing for Question Answering*, April, 2003.

Katz, B., Hurwitz, R., Lin, J. and Uzuner, O., "Better Public Policy through Natural Language Information Access," in *Proceeding of the National Conference on Digital Government Research*, May 2003, Boston MA.

Ibrahim, A., Katz, B., and Lin, J. "Extracting Structural Paraphrases from Aligned Monolingual Corpora." in *Proceedings of the Second International Workshop on Paraphrasing (IWP-2003)*, July 2003, Sapporo, Japan.

Tellex, S., Katz, B., Lin, J., Marton, G. and Fernandes, A. "Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering." in *Proceedings of the 26th*

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2003), July 2003, Toronto, Canada.

Barker, K., Blythe, J., Borchardt, G., Chaudhri, V. K., Clark, P. E., Cohen, P., Fitzgerald, J., Forbus, K., Gil, Y., Katz, B., Kim, J., King, G., Mishra, S., Murray, K., Otstott, C., Porter, B., Schrag, R. C., Uribe, T., Usher., J., and Yeh, P. Z. "A Knowledge Acquisition Tool for Course of Action Analysis" in *Proceedings of the AAAI Fifteenth Innovative Applications of Artificial Intelligence Conference (IAAI-03)*, Acapulco, Mexico, 2003.

Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., and Karger, D. R. "What Makes a Good Answer? The Role of Context in Question Answering." in *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT-2003)*, September 2003, Zurich, Switzerland.

Katz, B. and Lin, J. "Organizing and Accessing a Comprehensive Knowledge Base Using the World Wide Web." in *Proceedings of the IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS'03)*, October 2003, Cambridge, MA.

Lin, J. and Katz, B. "Question Answering from the Web Using Knowledge Annotation and Knowledge Mining Techniques." in *Proceedings of Twelfth International Conference on Information and Knowledge Management (CIKM-2003)*, November 2003, New Orleans, Louisiana.

Uzuner, O., Davis, R., and Katz, B. "Using Empirical Methods for Evaluating Expression and Content Similarity." in *Proceedings of HICSS-37 minitrack on Information Retrieval and Digital Library Applications*, January 2004, Waikoloa, Hawaii.

Katz, B., Lin, J., Stauffer, C., and Grimson, E. "Answering Questions about Moving Objects in Videos." in *New Directions in Question Answering*, M. Maybury (ed.), MIT Press, 2004.

Katz, B., Felshin, S., Lin, J., and Marton, G. "Viewing the Web as a Virtual Database for Question Answering." in *New Directions in Question Answering*, M. Maybury (ed.), MIT Press, 2004.

Additional References

Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P., *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell*, Garland Publishing, Inc., New York, 1998.

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. and Watson, J. D., *Molecular Biology of the Cell*, Third Edition, Garland Publishing, Inc., New York, 1994.

Barker, K., Blythe, J., Borchardt, G., Chaudhri, V. K., Clark, P. E., Cohen, P., Fitzgerald, J., Forbus, K., Gil, Y., Katz, B., Kim, J., King, G., Mishra, S., Murray, K., Otstott, C., Porter, B., Schrag, R. C., Uribe, T., Usher., J., and Yeh, P. Z. "A Knowledge Acquisition Tool for Course of Action Analysis" in *Proceedings of the AAAI Fifteenth Innovative Applications of Artificial Intelligence Conference (IAAI-03)*, Acapulco, Mexico, 2003.

Fuchs, N. E., Schwertel, U. and Schwitter, R., *Attempto Controlled English (ACE) Language Manual, Version 3.0*, Technical Report 99.03, Department of Computer Science, University of Zurich, 1999.

Katz, B., "Using English for Indexing and Retrieving," *Proceedings of the 1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO '88)*, 1988.

Katz, B., "Using English for Indexing and Retrieving," in P. H. Winston and S. A. Shellard, editors, *Artificial Intelligence at MIT: Expanding Frontiers*, volume 1, MIT Press, Cambridge, MA, 1990.

Katz, B., "Annotating the World Wide Web using Natural Language," *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97)*, 1997.

Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Ibrahim, A., Lin, J., Marton, G., McFarland, A. J. and Temelkuran, B., "Omnibase: Uniform Access to Heterogeneous Data as a Component of a Natural Language Question Answering System," *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems* (NLDB 02), 2002.

Knight, K. and Luk, S., "Building a Large Knowledge Base for Machine Translation," *Proceedings of the American Association of Artificial Intelligence Conference AAAI-94*. Seattle, WA, 1994.

Lenat, D. B., Guha, R. V., Pittman, K. and Pratt, D., "Cyc: Towards Programs with Common Sense," *Communications of the ACM*, 33(8):30–49, 1990.

Novak, J. D. and D. B Gowin. *Learning How to Learn.* Cambridge University Press, New York and Cambridge, UK, 1984.

Novak, J. D. *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations.* Lawrence Erlbaum and Associates, Mawah, NJ, 1998.

Ogden, C. K. *Basic English, International Second Language*, Harcourt, Brace, and World, New York, 1968.

Paustian, T., *Microbiology Webbed Out*, online textbook maintained at http://www.bact.wisc.edu/microtextbook/, University of Wisconsin-Madison, 2000.

Quillian, M. R., "The Teachable Language Comprehender: A Simulation Program and Theory of Language," *Communications of the ACM* 12:8, 459--476, 1969.